Tampereen yliopisto

Pinja Mikkonen

LUONNOLLISTA KIELTÄ KÄSITTELEVIEN NEUROVERKKOJEN KEHITYSASKELEITA

Kolmen neuroverkon vertailu

Informaatioteknologian ja viestinnän tiedekunta
Kandidaattitutkielma
Joulukuu 2019

Tiivistelmä

Pinja Mikkonen: Luonnollista kieltä käsittelevien neuroverkkojen kehitysaskeleita
Kandidaattitutkielma
Tampereen yliopisto
Tietojenkäsittelytieteiden tutkinto-ohjelma
Joulukuu 2019

Vuonna 2017 julkaistu Transformer-neuroverkkoarkkitehtuuri on tehostanut luonnollista kieltä käsittelevien neuroverkkojen suorituskykytestituloksia, ja Transformer-tyyppiset neuroverkot kuten BERT, XLNet ja RoBERTa ovat rikkoneet tuloslistojen ennätyksiä. Näiden mallien arkkitehtuurien tai koulutusmetodien vertailu on kuitenkin vaikeaa, sillä koulutusmateriaalin määrän ja koulutusajan kasvattaminen parantavat neuroverkon suoriutumista, eivätkä suorituskykyvertailut ota neuroverkon kouluttamisessa käytettyjä resursseja huomioon. Vertailen BERT:in, XLNet:in ja RoBERTa:n arkkitehtuureja, koulutusmetodeja ja koulutusmateriaalien määrää ja pohdin sitä, mitä ne voivat kertoa meille neuroverkkojen kouluttamisesta.

Avainsanat: Koneoppiminen, syväoppiminen, neuroverkot, luonnollisen kielen käsittely, Transformer-neuroverkot, kielimallit

1 Johdanto

Tietokoneiden kielentuotto on omat haasteensa omaava tutkimuksenala. Luonnollinen kieli on jatkuvasti kehittyvää ja monitulkintaista, eikä sen monipuolisia merkityksen vivahteita ole helppo muuttaa muotoon, jota algoritmit ymmärtäisivät. Neuroverkkoja on käytetty luonnollisen kielen käsittelyssä jo pitkään, muun muassa rekursiivisten neuroverkkojen muodossa [Goldberg 2017, 3]. Neuroverkot ovat osoittautuneet tehokkaiksi luonnollisen kielen käsittelijöiksi, jotka voivat oppia esimerkiksi tuottamaan tekstiä tai vastaamaan tekstikappaleen pohjalta esitettyihin kysymyksiin.

Vuonna 2017 Google AI julkisti uudenlaisen neuroverkkoarkkitehtuurin nimeltä Transformer, joka oli nopeammin ja tehokkaammin koulutettavissa kuin aiemmat, rekursiopohjaiset mallit [Vaswani *et al.* 2017]. Vuonna 2018 julkistettiin Transformer-arkkitehtuuria käyttävä BERT-neuroverkko, joka saavutti state-of-the-art -tuloksia useissa suorituskykytesteissä kuten GLUE ja SQuAD [Devlin *et al.* 2019].

BERT:in asettamat ennätykset kuitenkin rikottiin ennen pitkää. Kesällä 2019 BERT joutui luovuttamaan kärkipaikkansa muille, tehokkaammille malleille, kun XLNet-niminen Transformer rikkoi BERT:in ennätykset. XLNet:in kehitelleet Yang *et al.* [2019] toteuttivat XLNet:in arkkitehtuurissa ja koulutusmetodeissa useita BERT:iä koskevia parannusehdotuksia.

XLNet ei kuitenkaan säilyttänyt kärkipaikkaansa pitkään, vaan BERT:in arkkitehtuuria ja koulutusmetodeja käyttävä RoBERTa rikkoi tämän suorituskykytulokset [Liu *et al.* 2019]. Samalla Liu *et al.* [2019] nostivat esiin kysymyksen siitä, oliko BERT vain alikoulutettu suhteessa tätä seuranneisiin malleihin. Sekä XLNet että RoBERTa on koulutettu huomattavasti BERT:iä pidemmällä koulutusajalla ja käyttäen suurempia määriä koulutusmateriaalia [Liu *et al.* 2019], mikä tekee neuroverkkojen arkkitehtuurien ja koulutusmetodien tehokkuuden vertailun hyvin vaikeaksi. Liu *et al.* pyrkivät RoBERTa:lla todistamaan BERT:in arkkitehtuurin kilpailukyvykkyyden.

Tässä tutkielmassa esittelen BERT:in, XLNet:in ja RoBERTa:n arkkitehtuureja ja koulutusta. Vertailen myös näille neuroverkoille syötetyn koulutusmateriaalin määriä, koulutusajan pituutta ja käytettyjä eräkokoja. BERT alisuoriutuu varsinkin sellaisissa tehtävissä, jotka hyötyvät suuresta määrästä koulutusmateriaalia tai vaativat pitkän matkan riippuvuuksien hallintaa. Argumentoin, että BERT:in koulutustavoitteet haittasivat varsinkin pitkän matkan riippuvuuksien oppimista. Tämä tutkielma on toteutettu kirjallisuus-

katsauksena BERT:in, XLNet:in ja RoBERTa:n julkaisututkielmiin ja muuhun alan kirjallisuuteen. Tavoitteeni on antaa yleistasoinen käsitys siitä, mitkä tekijät vaikuttavat neuroverkkojen suorituskykyyn, ja kuinka mallien kokoerot voivat olla merkittävämpi tekijä kuin arkkitehtuuri tai koulutustavoitteet.

Aloitan käymällä läpi neuroverkkoteknologian perusteita, ja siirryn sitten esittelemään lyhyesti Transformer-arkkitehtuurin tärkeimmät piirteet. Käsittelen BERT:in, XLNet:in ja RoBERTa:n arkkitehtuureja ja koulutusmetodeja, minkä jälkeen vertailen lyhyesti luonnollisen kielen käsittelyn suorituskykytestejä ja sitä, millaisia tuloksia BERT, XLNet ja RoBERTa suorituskykytesteistä saivat. Lopuksi teen yhteenvedon saamistani tuloksista ja pohdin neuroverkkoteknologian tulevaisuudennäkymiä.

2 Luonnollista kieltä käsittelevät neuroverkot

Goldberg [2017, 1] määrittelee luonnollisen kielen käsittelyn ”luonnollista kieltä syötteenä ottavien tai sitä tuottavien algoritmien tai metodien suunnitteluksi”. Roskapostisuodattimet, automaattinen tekstinsyöttö tai virtuaaliset avustajat ovat kaikki esimerkkejä järjestelmistä jotka käsittelevät luonnollista kieltä. Luonnollisen kielen käsittelyn alakäsitteitä ovat esimerkiksi luonnollisen kielen ymmärtäminen (engl. *natural language understanding*, NLU), luonnollisen kielen tuottaminen (engl. *natural language generation*, NLG) ja kielellinen päättely (engl. *natural language inference*, NLI) [Li & Deng 2018; Wang *et al.* 2018]. Luonnollista kieltä ymmärtävät neuroverkot pystyvät esimerkiksi luokittelemaan tekstejä niiden aihepiirien perusteella tai päättelemään tekstien edustamia tunnetiloja [Wang *et al.* 2018]. Luonnollista kieltä tuottavat neuroverkot pystyvät generoimaan ymmärrettäviä luonnollisen kielen lauseita [Li *et al.* 2018, 329]. Kielelliseen päättelyyn pystyvät neuroverkot pystyvät arvioimaan lauseiden välisiä implikaatioita, kuten implikoiko lause ”En tiennyt, että X on totta” lausetta ”X” [Wang *et al.* 2018].

Monet neuroverkot opetetaan käsittelemään luonnollista kieltä kielimallin avulla (engl. *language model*, LM). Kielimalli on projektio siitä, mikä tietyn lauseen todennäköisyys on, tai vaihtoehtoisesti mikä on todennäköisin lauseessa seuraavaksi esiintyvä sana. [Goldberg 2017, 105.] Kielimalli voidaan kouluttaa joko valvotulla oppimisella (engl. *supervised learning*) tai valvomattomalla oppimisella (engl. *unsupervised learning*). Valvotussa oppimisessa neuroverkko tekee päätelmiä kielestä malliesimerkkeinä saamiensa syöte-tulosteparien pohjalta [Goldberg 2017, 13]. Valvotun oppimisen ongelmaksi muodostuu usein malliesimerkkien rajallinen määrä, joten kielimalleja koulutettaessa käännytään valvomattoman oppimisen puoleen.

Valvomattomassa oppimisessa neuroverkon parametrit alustetaan satunnaisesti, minkä jälkeen sille syötetään suuria määriä raakaa tekstiä. Tähän raakaan tekstiin voidaan generoida dynaamisesti malliesimerkkejä, kuten lauseen seuraavan sanan ennustamista. [Goldberg 2017, 115-116.] Neuroverkon parametrit päivitetään joka k:nnen koulutusesimerkin jälkeen, ja k:sta koulutusesimerkistä koostuvaa joukkoa kutsutaan koulutuseräksi (engl. *minibatch*) [Goldberg 2017, 61]. Valvomattomalla oppimisella esikoulutetut (engl. *pre-training*) neuroverkot voidaan jatkokouluttaa suorittamaan erikoistuneempia tehtäviä valvotulla oppimisella. Tätä vaihetta kutsutaan neuroverkon hienosäädöksi (*engl. fine-tuning*), ja sen aikana tehtävää A varten koulutettu neuroverkko koulutetaan suorittamaan tehtävää B. Hienosäädössä kaikki neuroverkon esikoulutuksessa oppimat parametrit jatkokoulutetaan sopimaan paremmin tehtävän B suoritukseen. [Devlin *et al.* 2019.] Valvomattomalla oppimisella suoritetun esikoulutuksen on todettu parantavan kielimallien suoriutumista ja vähentävän tarvetta testikohtaisille arkkitehtuureille [Devlin *et al.* 2018; Peters *et al.* 2018; Radford *et al.* 2018].

Neuroverkkojen kykyä käsitellä luonnollista kieltä ja neuroverkkoteknologian kehitystä mitataan muun muassa suorituskykytestien avulla. Suorituskykytestit koostuvat usein valmiiksi valikoidusta, testikohtaisesta koulutusmateriaalista ja standardisoidusta testistä. Suorituskykytestit tarjoavat useita erilaisia tehtäviä neuroverkoille, ja kuka tahansa voi ladata suorituskykytestin koulutusmateriaalin ja tarkastella neuroverkkonsa suoriutumista. Monet suurimmat suorituskykytestit ylläpitävät julkisia tuloslistoja niistä neuroverkoista, jotka saavuttavat testissä parhaat tulokset. [GLUE; ks. SQuAD & RACE.] Tutustumme eri suorituskykytesteihin tarkemmin luvussa 5.

3 Transformer-neuroverkkoarkkitehtuuri

Transformer-neuroverkkoarkkitehtuuri esiteltiin Vaswani *et al*:in [2017] tutkielmassa *Attention is all you need*. Transformer-arkkitehtuuri pohjautuu koodaaja-koodinpurkaja-rakenteeseen ja tarkkaavaisuusmekanismeihin, mikä tekee Transformerista nopeammin koulutettavan ja mahdollistaa laskennan rinnakkaistamisen paremmin kuin aiemmat neuroverkkoarkkitehtuurit [Vaswani *et al.* 2017].

Koodaaja-koodinpurkaja-arkkitehtuuri (engl. *encoder-decoder*) on NLP-neuroverkkoarkkitehtuuri, jossa koodaaja muuntaa saamansa syötteen vektoriksi, jonka pohjalta koodinpurkaja puolestaan muodostaa tekstimuotoisen tulosteen. Koodaaja-koodinpurkaja-arkkitehtuureja käytettiin alun perin konekääntämiseen. [Cho *et al.* 2014.] Tarkkaavaisuusmekanismi (engl. a*ttention*) kehitettiin tukemaan koodaaja-koodinpurkaja-arkkitehtuuria pitkien lauseiden kääntämisessä. Tarkkaavaisuusmekanismissa koodinpurkaja muodostaa useita vektoreita lauseen eri sanoille. Nämä vektorit painotetaan suhteessa toisiinsa sen mukaan, miten paljon kontekstia ne antavat lauseen muille sanoille. Tämä

antaa koodinpurkajan hyödyntää koodaajan vektoreita tarpeen mukaan, ja parantaa pidempien lauseiden käännettävyyttä. Tällä myös vältytään siltä, että koodaajan pitää tallentaa lähdelauseen kaikki relevantti informaatio yhteen vektoriin. [Bahdanau *et al.* 2014.]

Transformerin arkkitehtuuri koostuu kuudesta identtisestä koodaaja-koodinpurkakerroksesta ja käyttää monipäistä tarkkaavaisuusmekanismia. Monipäinen tarkkaavaisuusmekanismi suorittaa tarkkaavaisuusfunktion usealle eri kyselylle yhtä aikaa, mikä antaa mallin käsitellä informaatiota useista eri lauseen sijainneista. Tämä poistaa tarpeen rekursiolle arkkitehtuurissa. [Vaswani *et al.* 2017] Transformerin arkkitehtuuri mahdollisti luonnollisen kielen ei-lineaarisen käsittelyn, sillä monipäisen tarkkaavaisuusmekanismin ansiosta koodinpurkaja pystyi kiinnittämään lauseessa huomiota kohtiin, jotka tarjosivat eniten kontekstia kulloinkin käsitellylle sanalle.

4 Neuroverkot

Tässä kappaleessa esittelen BERT:in, XLNet:in ja RoBERTa:n. Käyn neuroverkot läpi niiden julkaisuajankohtien mukaisessa järjestyksessä, aloittaen BERT:istä ja siirtyen sitten XLNet:in kautta RoBERTa:an. XLNet:in toteutukseen on toteutettu piirteitä Transformer-XL-neuroverkosta, joka esitellään lyhyesti XLNet:in yhteydessä.

4.1 BERT

BERT (Bidirectional Encoder Representations from Transformers) oli Transformer-arkkitehtuurilla toteutettu kielenmallinnusjärjestelmä, joka rikkoi useiden suorituskykytestien aiemmat ennätykset [Devlin *et al.* 2018]. BERT käytti *Attention is all you need* -artikkelissa esiteltyä Transformer-arkkitehtuuria, mutta esitteli myös kaksisuuntaisen kielimallin koulutusmetodin, jonka Devlin *et al.* [2018] arvioivat olevan yksisuuntaisia kielimalleja tehokkaampi ja hyödyntävän kielimallin esikoulutuksen koko potentiaalia. Kaksisuuntaisella koulutuksella tarkoitetaan esikoulutusta, jossa neuroverkolle annetaan tehtäväksi ennustaa satunnainen peitetty sana tarkastelemalla lauseen kaikkia muita sanoja.

BERT:iä edeltävät, suorituskykyvertailuja hallinneet neuroverkot kouluttivat kielimallinsa käyttäen joko vasemmalta oikealle suuntautuvaa kontekstia, kuten GPT [Radford *et al.* 2018], tai kouluttamalla erilliset mallit sekä oikealta vasemmalle suuntautuvalle kontekstille että vasemmalta oikealle suuntautuvat kontekstille, kuten ELMo [Peters *et al.* 2018]. Yksisuuntaisesti koulutettu kielimalli on koulutettu ennustamaan lauseessa seuraavaksi todennäköisimmin esiintyvä sana tarkastelemalla lauseen aiempia sanoja [Goldberg 2017, 105]. Devlin *et al.* [2018] argumentoivat tämän olevan tehotonta ja estävän kielimallia saavuttamasta täyttä potentiaaliaan, sillä useat NLP-tehtävät vaativat

kykyä käyttää kaksisuuntaista kontekstia, kuten esimerkiksi kysymyksiin vastaaminen tai päättelytehtävät.

BERT:in arvioitiin myös soveltuvan useiden erilaisten tehtävien suorittamiseen ilman tarvetta erilliselle arkkitehtuurille jokaista tavoitetta varten. Kaksisuuntaisesti koulutetun kielimallin arvioitiin pystyvän oppimaan kieltä monipuolisesti, ja näin ollen hyödyntämään mahdollisimman suuren osan esikoulutuksessa laadituista parametreista. BERT ei siis vaatisi kuin hienosäädön suoriutuakseen erilaisista NLP-tavoitteista. [Devlin *et al.* 2018.]

BERT:in kaksisuuntainen kielimalli nimettiin naamioiduksi kielimalliksi (engl. *masked language model,* MLM). Naamioitu kielimalli koulutetaan syöttämällä tälle suuria määriä raakaa tekstiä ja korvaamalla osa syötteen sanoista erikoismerkillä `[MASK]`. Kielimallin tehtävä on päätellä tekstin muiden sanojen perusteella, mikä sana erikoismerkin tilalle kuuluu. Yhteensä 15 % koulutusmateriaalin sanoista naamioidaan. Näistä naamioiduista sanoista 80 % korvataan erikoismerkillä `[MASK]`, 10 % satunnaisella erikoismerkillä ja 10 % kerroista naamiona toimii sana itse. Tämän on tarkoitus pienentää ristiriitaa esikoulutuksen ja hienosäädön välillä, sillä erikoismerkki `[MASK]` ei esiinny materiaalissa esikoulutuksen jälkeen, ja tämän pelättiin vaikuttavan negatiivisesti BERT:in suoriutumiseen. [Devlin *et al.* 2018.]

BERT pyrittiin opettamaan myös ymmärtämään kahden tekstikatkelman välisiä riippuvuuksia. Kahden eri tekstikatkelman välisen suhteen ymmärtäminen on tärkeää esimerkiksi kysymyksiin vastaamisessa tai kielellisissä päättelytehtävissä, jotka vaativat kahden eri lauseen välisen suhteen ymmärtämistä. Kielimallit ovat perusluonteeltaan heikkoja oppimaan lauseiden välisiä riippuvuuksia, joten BERT:in suoritusta pyrittiin tukemaan antamalla tälle koulutustehtäväksi seuraavan lauseen ennustustavoite (engl. *next sentence prediction*, NSP). Seuraavan lauseen ennustaminen suoritettiin syöttämällä BERT:ille kaksi tekstikatkelmaa, joiden yhteispituus oli alle 512 sanaa. Mikäli tekstinkatkelmat olivat peräisin samasta lähteestä ja seurasivat toisiaan, niistä jälkimmäiselle annettiin tunniste OnSeuraava. Muulloin tunniste oli EiOleSeuraava. BERT joutui päättelemään jälkimmäisen lauseen tunnisteen. [Devlin *et al.* 2018.]

BERT:istä koulutettiin kaksi erikokoista mallia, 110M parametria sisältänyt BERTbase ja 340M parametria sisältänyt BERTlarge. BERTlarge pärjäsi kaikissa suorituksissa paremmin kuin BERTbase, mikä Devlin *et al*:in [2018] mielestä osoitti, että neuroverkon mallin koon kasvattaminen parantaa näiden suoritusta esikoulutuksen koulutusmateriaalin määrästä riippumatta. Esikoulutusajan nopeuttamiseksi BERT koulutettiin yhteensä 128 sanan pituisilla tekstikatkelmilla ensimmäiset 90 % koulutusajasta, minkä jälkeen tekstikatkelmien pituus kasvatettiin 512 sanaan. Molemmat mallit koulutettiin 16GB koulutusmateriaalia ja 256 kokoisilla koulutuserillä. [Devlin *et al.* 2018.]

4.2 XLNet

XLNet on kesällä 2019 julkaistu, Transformer-arkkitehtuuria käyttävä kielimalli, joka pyrkii yhdistämään yksi- ja kaksisuuntaisten kielimallien parhaat puolet hyödyntämällä permutaatiopohjaista esikoulutusta [Yang *et al.* 2019]. BERT-tyyppisellä kielimallin koulutusmetodilla on monia vahvuuksia, mukaan lukien tämän tarjoama kaksisuuntainen konteksti, mutta siinä on Yang *et al.* [2019] mukaan myös heikkouksia.

Yksi BERT:in ongelmista on sen naamiointimekanismin aiheuttama esikoulutus-hienosäätökonflikti. Kyseinen konflikti huomioitiin jo BERT:in koulutuksessa, ja vaihtelu naamiona käytettyjen merkkien välillä oli yritys ehkäistä tätä [Devlin *et al.* 2018]. Yang *et al.* [2019] huomauttivat myös, että BERT ei ota huomioon naamioitujen sanojen välistä potentiaalista kontekstia. Mikäli BERT:ille tarjotussa esikoulutussyötteessä naamioidaan kaksi toisilleen kontekstia tuottavaa sanaa, BERT tekee ennusteen kullekin sanalle erikseen ottamatta toisen tarjoamaa kontekstia huomioon. Esimerkiksi mikäli lauseessa ”Matkustan New Yorkiin” sekä sanat ”New” että ”York” korvataan `[MASK]` -symbolilla, niin BERT ei ota sanoja ennustaessaan huomioon, että nämä sanat voisivat vaikuttaa toinen toisiinsa. [Yang *et al.* 2019.]

XLNet pyrki korjaamaan edellä mainitut esikoulutusdatan korruptoinnin ja itsenäisyysoletuksen. XLNet:in kielimallin koulutusmetodi on yritys yhdistää yksi- ja kaksisuuntaisten kielimallien parhaat puolet, ja säilyttää lauseiden molemminpuolinen konteksti ilman tarvetta syötteen korruptoimiselle. XLNet käyttää permutaatiopohjaista kielimallia, jonka esikoulutuksen aikana tekstikatkelmien sanat syötetään XLNet:ille niiden kaikissa mahdollisissa järjestyksissä. Näin XLNet oppii tekemään ennusteita ottaen lauseen molemminpuoliset kontekstit huomioon, mutta ei kohtaa esikoulutus-hienosäätökonflikteja korruptoituneen syötteen takia. [Yang *et al.* 2019.]

XLNet:in arkkitehtuuri omaksui piirteitä Transformer-XL:stä, joka pyrki tehostamaan Transformerien pitkän matkan riippuvuuksien oppimista [Dai *et al.* 2019]. Ensinnäkin XLNet soveltaa Transformer-XL:n relatiivista sijaintikoodausta, joka tallettaa sanojen sijainnit ainoastaan suhteessa muihin saman tekstikatkelmien sanoihin. Tämä auttaa XLNet:in permutaatiopohjaista kielimallia omaksumaan lauserakenteita. Toinen Transformer-XL:stä omaksuttu piirre on Transformer-XL:n muistina toimiva, tekstikappaletasoinen rekursiomekanismi. [Yang *et al.* 2019.] Transformer-XL kierrättää aiempien tekstikatkelmien laskelmia, mikä toimii muistina kulloinkin käsiteltävänä olevalle osiolle ja rakentaa rekursiota eri osien väliin. Tämä auttaa Transformer-XL:ää hyödyntämään pidemmän matkan kontekstia kielimallia koulutettaessa ja ehkäisee niin sanotun ”kontekstihajoamisen” ongelmaa, jossa konteksti katoaa kahden eri tekstikatkelman välissä. [Dai *et al.* 2019.] XLNet:iin pyrittiin toteuttamaan vastaavanlainen pidemmän konteks-

tin omaksuminen, ja XLNet suoriutuikin hyvin pitkän matkan kontekstia vaativissa suorituskykytesteissä kuten RACE:ssa [Yang *et al.* 2019].

XLNet esikoulutettiin 126GB:llä raakaa tekstiä, ja sen mallin koko pidettiin BERTlarge:a vastaavana vertailun helpottamiseksi. XLNet:iä koulutettiin yhteensä 512 sanan pituisilla tekstikatkelmilla ja 2048 kokoisilla koulutuserillä. [Yang *et al.* 2019.]

4.3 RoBERTa

RoBERTa (Robustly optimized BERT approach) vastasi suureen osaan BERT:iin kohdistetusta kritiikistä. Liu *et al.* [2019] argumentoivat BERT:in olevan alikoulutettu, ja että laajemmalla esikoulutuksella ja suuremmalla määrällä parametreja BERT-tyylinen kielimallin koulutus olisi kilpailukykyistä.

RoBERTa käytti samaa arkkitehtuuria kuin BERT, mutta teki muutoksia BERT:in koulutusmetodeihin. RoBERTa:a koulutettiin pidempään ja isommilla koulutuserillä kuin BERT:iä. BERT:iin sisällytetty seuraavan lauseen ennustus -koulutustehtävä poistettiin, sillä sen tehokkuus lauseiden välisten suhteiden opetuksessa kyseenalaistui. RoBERTa:a koulutettiin myös pidemmillä lauseilla kuin BERT:iä, ja sen naamiointimalli muutettiin dynaamiseksi, jotta esikoulutus-hienosäätökonfliktia ei tapahtuisi. BERT:in naamiointi oli staattista, eli naamiot generoitiin kerran esiprosessoinnin aikana. Koulutusmateriaali syötettiin BERT:ille useita kertoja, joten koulutusmateriaalille generoitiin kymmenen eri naamiointimallia. Tästä huolimatta BERT kohtasi kunkin tekstikatkelman naamioituna samalla tavalla neljä kertaa. RoBERTa generoi eri naamiointimallin joka kerta, kun tekstikatkelma syötettiin kielimallille. RoBERTa myös koulutettiin BERT:iä isommalla määrällä koulutusmateriaalia. Raakatekstiä käyttävä valvomaton oppiminen hyötyy todistetusti suuresta määrästä esikoulutustekstiä, minkä takia RoBERTa:n koulutusmateriaalin määrä kymmenkertaistettiin. [Liu *et al.* 2019.]

RoBERTa:n koulutuksen aikana tutkittiin, miten paljon BERT:in eri koulutuselementit vaikuttivat tämän suoriutumiseen. RoBERTa:n dynaamisen naamioinnin tulokset olivat lievästi parempia kuin BERT:in staattisen. Liu *et al.* [2019] tutkivat myös BERT:in NSP-tavoitteen onnistuneisuutta. Kuten luvussa 4.1 kerroimme, BERT oltiin koulutettu tunnistamaan toisiaan seuraavia tekstikatkelmia. RoBERTa tutki sitä, vaikuttiko tämä BERT:in suoriutumiseen positiivisesti. Tutkimuksen johtopäätös oli, että RoBERTa pärjäsi paremmin kun tälle syötettiin pidempiä, yhtenäisiä tekstikatkelmia ilman seuraavan lauseen ennustustavoitetta. Tulokset olivat parhaita silloin, kun tekstikatkelmat oltiin valittu vain yhdestä dokumentista, mutta tämä sai aikaan liikaa eroja koulutuserien pituuksissa. Tämän välttämiseksi RoBERTa koulutettiin täyspituisilla koulutuserillä, jotka saivat ylittää dokumenttien rajoja. [Liu et al. 2019.]

Liu et al. [2019] tarkastelivat myös esikoulutuksessa käytetyn koulutusmateriaalin määrän ja läpikäyntikertojen vaikutusta kielimallin suoriutumiseen, ja totesivat näiden kahden tekijän vaikuttavan RoBERTa:n suoriutumiseen positiivisesti. Tulokset tarjosivat uuden selityksen sille, miksi XLNet suoriutui dramaattisesti BERT:iä paremmin. Tutkijat huomauttivat, että XLNet koulutettiin kymmenen kertaa isommalla määrällä koulutusmateriaalia kuin BERT, ja kahdeksan kertaa suuremmilla koulutuserillä joissa oli puolet vähemmän optimointiaskeleita. Tämä tarkoittaa sitä, että XLNet näki neljä kertaa enemmän merkkijonoja kuin BERT koulutuksensa aikana. RoBERTa:lla suoritetut jatkotutkimukset osoittivat, että pidemmät koulutusajat ja suuremmat koulutusmateriaalin määrät aiheuttivat kasvavan positiivisen vaikutuksen RoBERTa:n suoriutumiseen. [Liu et al. 2019.]

RoBERTa:sta koulutettiin useita eri kokoisia versioita vertailun vuoksi, mutta suorituskykyvertailuissa käytettiin suurinta mallia. Tässä mallissa oli 355 miljoonaa parametria, mikä vastasi BERTlargea. Koulutus tapahtui 160GB:llä koulutusmateriaalia käyttäen eräkokoa 8000 ja 512 sanan pituisia tekstikatkelmia. [Liu *et al.* 2019.]

5 Suorituskykytestien tulokset

Tämän osion vertailut on suoritettu alkuperäisten julkaisumateriaalien perusteella. Keskitän tarkasteluni vain yksittäismallien tuloksiin, sillä vaikka usean eri neuroverkkomallin keskivertovastausten käyttäminen parantaakin niiden suoritusta, se myös hankaloittaa mallien suoraa vertailua keskenään. BERT:in kohdalla käytän BERTlarge:n tuloksia, sillä BERTlarge vastaa paremmin XLNet:in ja RoBERTa:n kokoa.

Käsittelen vain sellaisia suorituskykytestejä, jotka kaikki neuroverkot ovat suorittaneet; näitä ovat GLUE, SQuAD v1.1 ja v2.0 sekä RACE. Aloitan kertomalla eri suorituskykytesteistä, minkä jälkeen siirryn analysoimaan neuroverkkojen saamia tuloksia ja niihin mahdollisesti vaikuttavia tekijöitä.

5.1 Suorituskykytestit lyhyesti

Suorituskykytestejä käytetään mittaamaan standardisoidusti neuroverkkojen kykyä käsitellä luonnollista kieltä. Suorituskykytestit koostuvat yhdestä tai useammasta testistä, jotka voi ladata suorituskykytestin verkkosivuilta ja kyseisen testin koulutusmateriaalista. Testien tulokset antavat arvion mallin suoriutumisesta eri luonnollisen kielen käsittelyn tehtävissä. [GLUE; SQuAD; RACE]

GLUE-suorituskykytesti (General Language Understanding Evaluation) on yhdeksän eri suorituskykytestin kokoelma, joka mittaa luonnollisen kielen ymmärrystä [Wang *et al.* 2018]. GLUE:n eri testit esitellään taulukossa 1. GLUE on suosittu, sillä se testaa monipuolisesti erilaisia luonnollisen kielen ymmärtämisen osa-alueita, ja ylläpitää julkista tu-

lostaulukkoa. Jätän WNLI-suorituskykytestin pois tarkastelustani, sillä BERT ja XLNet eivät suorittaneet kyseistä testiä yksittäismalleilla.

Testin nimi	Testin luokitus	Testin tavoite
CoLA (The Corpus of Linguistic Acceptability)	Yhden lauseen testit	Lauseen kieliopillisen korrektiuden arviointi
SST-2 (Stanford Sentiment Treebank)	Yhden lauseen testit	Lauseen edustaman tunnetilan päättely akselilla positiivinen-negatiivinen
MRPC (Microsoft Research Paraphrase Corpus)	Samanlaisuus- ja kiertoilmaustestit	Kahden lauseen semanttisen samanlaisuuden arviointi
QQP (Quora Question Pairs)	Samanlaisuus- ja kiertoilmaustestit	Kahden kysymyksen semanttisen samanlaisuuden arviointi
STS-B (Semantic Textual Similarity Benchmark)	Samanlaisuus- ja kiertoilmaustestit	Kahden lauseen semanttisen samanlaisuuden arviointi asteikolla 1-5
MNLI (Multi-Genre Natural Language Inference Corpus)	Päättelytestit	Kahden lauseen luokittelu toisensa implikoiviksi, ristiriitaisiksi tai neutraaleiksi
QNLI (Question-answering NLI)	Päättelytestit	Päättele löytyykö vastaus kysymykseen annetusta tekstikappaleesta
RTE (Recognizing Textual Entailment)	Päättelytestit	Lauseiden välisen ristiriidan tai implikaation päättely
WNLI (Winograd NLI)	Päättelytestit	Pronominin osoittaman subjektin löytäminen lauseesta

Taulukko 1: GLUE:n suorituskykytestit [Wang *et al.* 2018]

SQuAD v1.1 ja v2.0 tarkoittavat Stanford Question Answering Dataset:iä ja ovat luetunymmärtämistä testaavia suorituskykytestejä. Testeissä neuroverkolle annetaan syötteenä Wikipedia-artikkeli ja kyseistä artikkelia koskeva kysymys. Testiä suorittavan neuroverkon tehtävä on osoittaa artikkelista kohta, joka vastaa kysymykseen. [Rajpurkar *et al.* 2016.] SQuAD v1.1:ssä vastaus löytyy annetusta artikkelista joka kerta, mutta SQuAD

v2.0:ssa on mahdollista että artikkeli ei sisällä vastausta kysymykseen [Rajpurkar *et al.* 2016; Rajpurkar *et al.* 2018].

RACE (ReAding Comprehension dataset collected from English Examinations) koostuu kiinalaisille 12–18-vuotiaille opiskelijoille suunnatuista englannin kielen luetunymmärtämistehtävistä. Testi jakautuu kahteen eri osioon: yläasteikäisten kysymyksiin ja lukioikäisten kysymyksiin. RACE:ssa neuroverkolle annetaan tekstikappale ja useita kappaleeseen liittyviä kysymyksiä, joihin on neljä eri vastausvaihtoehtoa. [Lai *et al.* 2017.] RACE on tunnettu haastavuudestaan, sillä monet vastaukset ovat tulkinnanvaraisia ja tekstikappaleet ovat usein pitkiä. Monet kysymykset vaativat myös päättelykykyä. [Lai *et al.* 2017; RACE.]

5.2 Suorituskykytestien tulokset

Neuroverkko	SQuAD v1.1 (EM / F1*)	SQuAD v2.0 (EM / F1*)	RACE (Acc / Middle / High**)
BERT	85.1 / 91.8	80.0 / 83.1	72.0 / 76.6 / 70.1
XLNet	**89.0** / 94.5	86.1 / 88.8	81.8 / 85.5 / 80.2
RoBERTa	88.9 / **94.6**	**86.5** / **89.4**	**83.2** / **86.5** / **81.3**

Taulukko 2: SQuAD v1.1, v2.0 ja RACE -testien tulokset [Devlin *et al.* 2018; Yang *et al.* 2019; Liu *et al.* 2019; Pan *et al.* 2019.]

* EM = Oikea vastaus, F1 = Keskimääräinen vastauksen ja oikean vastauksen päällekkäisyys (%) [Rajpurkar *et al.* 2016.]

** Acc = Oikeita vastauksia, Middle = Yläaste-kategoria, High = Lukio-kategoria (%)

Taulukosta 2 käy ilmi, että RoBERTa saavuttaa parhaat tulokset RACE:ssa ja SQuAD v2.0:ssa. Tämän lisäksi RoBERTa:n ja XLNet:in SQuAD v1.1 -suoritukset ovat likipitäen samat. Tämä vahvistaa Liu *et al*:in [2019] arviota siitä, että koulutusmateriaalin määrän ja koulutuserien koon kaltaiset tekijät ovat tärkeämpiä kuin arkkitehtuuri tai kielimallin koulutusmetodi.

BERT:in RACE-testin tulokset ovat selkeästi XLNet:in ja RoBERTa:n tuloksia alhaisemmat. RACE on vaikea suorituskykytesti, joka testaa kielen pitkän matkan riippuvuuksien hahmottamista [RACE]. Kuten taulukko 2 näyttää, XLNet saavutti julkaisunsa yhteydessä +7.5 % parannuksen RACE:n aiempaan kärkisijan pitäjään BERT:iin verrattuna. XLNet:in pohjana käytetty Transformer-XL on suunniteltu oppimaan tehokkaasti pitkän matkan riippuvuuksia [Dai *et al.* 2019], ja Yang *et al.* [2019] epäilivät alun perin tämän vaikuttaneen positiivisesti XLNet:in saavuttamaan RACE-tulokseen. RoBERTa kuitenkin ylittää XLNet:in saaman tuloksen ilman Transformer-XL:n arkkitehtuurista omaksuttuja piirteitä tai muita pitkän matkan riippuvuuksia tukevia piirteitä [Liu *et al.* 2019]. Tämä kyseenalaistaa Yang *et al*:in [2019] arvion todennäköisyyden, ja on todennäköisempää, että BERT on koulutuksensa takia alisuoriutunut RACE:ssa. BERT:in saa-

mat syötteet olivat seuraavan lauseen ennustustavoitteen takia 50 % ajasta peräisin kahdesta eri lähteestä, minkä lisäksi 90 % BERT:in koulutuksesta tapahtui lyhennetyillä tekstisarjoilla koulutusajan lyhentämiseksi [Devlin *et al.* 2019]. RoBERTa ja XLNet koulutettiin täyspituisilla tekstisarjoilla, jotka olivat pääosin peräisin yhdestä lähteestä [Yang *et al.* 2019; Liu *et al.* 2019]. On mahdollista, että lyhennetyt tekstikappaleet ja seuraavan lauseen ennustustavoite ovat saaneet BERT:in alisuoriutumaan varsinkin RACE:n kaltaisissa tehtävissä, jotka vaativat pitkän kontekstin hahmottamista.

Liu *et al.* [2019] huomauttavat myös käyttäneensä RoBERTa:n SQuAD-koulutuksessa vain SQuAD:in koulutusmateriaalia, toisin kuin XLNet ja BERT, jotka käyttivät sallittuja lisämateriaaleja [Devlin *et al.* 2018; Yang *et al.* 2019]. Tämä tarkoittaa sitä, että käyttämällä hienosäädössä lisämateriaalia RoBERTa:n SQuAD -suoritus saattaa parantua.

Neuro-verkko	CoLA	SST-2	MRPC	STS-B	QQP (Acc)	MNLI-m/-mm*	QNLI	RTE
BERTlarge	60.5	94.9	89.3	86.5	89.3	86.7/85.9	92.7	70.1
XLNet	63.6	95.6	89.2	91.8	91.8	89.8/-	93.9	83.8
RoBERTa	**68.0**	**96.4**	**90.9**	**92.4**	**92.2**	**90.2/90.2**	**94.7**	**86.6**

Taulukko 3: GLUE:n testitulokset (%) [Devlin *et al.* 2019; Yang *et al.* 2019; Liu *et al.* 2019.]
* m/mm = Samaa / eri tyylilajeja edustavat tekstit (engl. *(mis)matched*) [Wang *et al.* 2018.]

Taulukosta 3 näkyy, että RoBERTa on saavuttanut parhaat tulokset myös GLUE:n suorituskykytesteissä. Tämä yhdessä taulukon 2 tulosten kanssa vahvistaa hypoteesin siitä, että valvomattomalla oppimisella ja tarpeellisilla koulutusmäärillä koulutettu kielimalli voi suoriutua hyvin monissa eri luonnollisen kielen ymmärtämisen tavoitteissa.

RoBERTa osoittaa parannusta CoLA-suorituskykytestissä suhteessa sekä BERT:iin että XLNet:iin, mutta kaikki kolme neuroverkkoa alisuoriutuvat kyseisessä testissä suhteessa valtaosaan muista GLUE:n tuloksista. Wang *et al.* [2018] mukaan valvotulla oppimisella CoLA:a varten koulutetut mallit suoriutuvat CoLA:ssa paremmin kuin valvomattomalla oppimisella koulutetut. Tämä selittää tarkastelemieni kolmen mallin alisuoriutumisen CoLA:ssa muihin GLUE:n testeihin nähden. RoBERTa:n aikaansaama parannus voi olla myös merkki siitä, että suurempi määrä koulutusmateriaalia voi parantaa myös valvomattomalla oppimisella koulutettujen mallien suoritusta myös kielioppia testaavissa tehtävissä.

BERT alisuoriutuu selvästi RTE-testissä. RTE testaa lausetason implikaatioiden hahmottamista, ja tarjoaa paljon vähemmän koulutusmateriaalia kuin muut päättelytehtävät QNLI ja MNLI. RTE:n koulutusmateriaalissa on 2 500 koulutusesimerkkiä, siinä missä MNLI ja QNLI tarjoavat 393 000 ja 105 000 koulutusesimerkkiä [Wang *et al.* 2018].

Kuten luvussa 2 totesin, on tutkitusti todistettu, että laaja valvomaton oppiminen parantaa kielimallien suoriutumista myös niissä suorituskykytesteissä, joihin ei ole olemassa suuria määriä koulutusmateriaalia [Goldberg 2017, 115–116]. RoBERTa ja XLNet koulutettiin kymmenen kertaa isommalla määrällä esikoulutusmateriaalia kuin BERT [Liu *et al.* 2019], mikä näkyy näiden paremmissa RTE-suorituksissa. RoBERTa myös koulutti RTE-mallinsa hienosäädetyn MNLI-mallin pohjalta, mikä on mahdollisesti parantanut tämän tulosta [Liu *et al.* 2019].

6 Yhteenveto

Luvun 5 tulokset kertovat monia asioita neuroverkkojen koulutuksesta. Suuri määrä koulutusmateriaalia ja suurempi koulutuserien koko saavat aikaan parempia koulutustuloksia, samoin pidempi koulutusaika. Kielimallin laajempi koulutus vaikuttaa vahvistavan kielimallin suoriutumista myös sellaisissa tehtävissä, joihin on käsillä vähän koulutusmateriaalia. Luvun 5 tulokset viittaavat siihen, että kielimallin kouluttaminen pitkillä, yhtenäisillä tekstikatkelmilla parantaa tämän suoritusta kielellistä päättelyä vaativissa suorituskykytesteissä, mutta tämän tutkielman pienen otannan puolesta aiheesta pitää tehdä jatkotutkimusta.

Mallien arkkitehtuurien tai koulutustavoitteiden merkitystä on kuitenkin hankala arvioida eroavien koulutusmateriaalin määrien ja mallien kokoerojen takia. Tämän takia suorituskykytestit eivät ole täysin puolueeton standardi: ne eivät ota tuloksissaan huomioon sitä, saavuttaako neuroverkko tuloksensa innovatiivisella arkkitehtuurilla vai suurella määrällä laskennallisia resursseja. Neuroverkkojen koulutus on myös käymässä kalliimmaksi ja vaatii yhä enemmän resursseja, mikä vaikeuttaa tieteellistä vertaisarviointia. Neuroverkkoyhteisössä onkin keskusteltu siitä, pitäisikö suorituskykytesteillä olla tarkemmat kokorajoitukset, jotta mallien arkkitehtuuri on paremmin vertailtavissa. Anna Rogers [2019] ehdottaakin artikkelissaan *How the Transformers Broke NLP Leaderboards* seuraavia muutoksia suorituskykylistoihin: joko suorituskykytestit alkavat rajoittaa sallitun koulutusmateriaalin kokoa, tai ottavat kulutetun koulutusajan ja käytetyn koulutusmateriaalin määrän huomioon tuloslistojen pisteytyksissä. Nämä tekijät estäisivät Rogersin [2019] mukaan suorituskykyvertailuja muuttumasta resurssikilpailuiksi.

Tämän tutkielman kirjoitusprosessin aikana neuroverkkoteknologia on ehtinyt ottaa jälleen uusia harppauksia. RoBERTa ja XLNet ovat tipahtaneet sijoille viisi ja kuusi GLUE-suorituskykytestissä [GLUE]. Neuroverkkoteknologia muuttuu nopeasti, mikä tarkoittaa että on aiempaa tärkeämpää saada vertaisarvioitavissa olevia ja vertailukelpoisia tuloksia. Muussa tapauksessa käytettävissä olevat resurssit saattavat olla tärkein neuroverkon suoriutumiseen vaikuttava tekijä.

7 Lähdeluettelo

Deep Learning in Natural Language Processing. 2018. Edited by Li Deng, Yang Liu. Singapore: Springer. *http://ebookcentral.proquest.com/lib/tampere/detail.action?docID=5401147*.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2014. *Neural Machine Translation by Jointly Learning to Align and Translate.* arXiv Preprint arXiv:1409.0473.

Cho, Kyunghyun, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*. arXiv Preprint arXiv:1406.1078.

Dai, Zihang, Zhilin Yang, Yiming Yang, William W. Cohen, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. 2019. *Transformer-Xl: Attentive Language Models Beyond a Fixed-Length Context*. arXiv Preprint arXiv:1901.02860.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. *Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding*. arXiv Preprint arXiv:1810.04805.

Goldberg, Yoav. 2017. *Neural Network Methods for Natural Language Processing*. Synthesis Lectures on Human Language Technologies 10 (1): 1-309.

Lai, Guokun and Xie, Qizhe and Liu, Hanxiao and Yang, Yiming and Hovy, Eduard. 2017 RACE: Large-Scale ReAding Comprehension Dataset from Examinations. Haettu 28.11.2019, http://www.qizhexie.com/data/RACE_leaderboard.html.

Lai, Guokun, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. *Race: Large-Scale Reading Comprehension Dataset from Examinations*. arXiv Preprint arXiv:1704.04683.

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. *Roberta: A Robustly Optimized Bert Pretraining Approach.* arXiv Preprint arXiv:1907.11692.

Pan, Xiaoman, Kai Sun, Dian Yu, Heng Ji, and Dong Yu. 2019. *Improving Question Answering with External Knowledge.* arXiv Preprint arXiv:1902.00993.

Peters, Matthew E., Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. *Deep Contextualized Word Representations.* arXiv Preprint arXiv:1802.05365.

Pranav Rajpurkar, Robin Jia, Percy Liang. 2019. *SQuAD 2.0: The Stanford Question Answering Dataset.* GitHub. Haettu 28.11.2019. https://rajpurkar.github.io/SQuAD-explorer/.

Radford, Alec, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. *Improving Language Understanding by Generative Pre-Training.* URL Https://S3-Us-West-2.Amazonaws.Com/Openai-Assets/Researchcovers/Languageunsupervised/Language Understanding Paper.Pdf.

Rajpurkar, Pranav, Robin Jia, and Percy Liang. 2018. *Know what You Don'T Know: Unanswerable Questions for SQuAD*. arXiv Preprint arXiv:1806.03822.

Rajpurkar, Pranav, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. *Squad: 100,000 Questions for Machine Comprehension of Text.* arXiv Preprint arXiv:1606.05250.

Rogers, Anna. 2019. *How the Transformers Broke NLP Leaderboards*. Hacking Semantics. Haettu 28.11.2019, https://hackingsemantics.xyz/2019/leaderboards/.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. *Attention is all You Need.*

Wang, Alex and Singh, Amanpreet and Michael, Julian and Hill, Felix and Levy, Omer and Bowman, Samuel R. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. Haettu 29.11.2019, https://gluebenchmark.com/leaderboard.

Wang, Alex, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. *Glue: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding.* arXiv Preprint arXiv:1804.07461.

Yang, Zhilin, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. *XLNet: Generalized Autoregressive Pretraining for Language Understanding*. arXiv Preprint arXiv:1906.08237.